PREDICTION OF ICD-9 CODE ASSIGNMENT USING ATTENTION-BASED CONVOLUTIONAL NEURAL NETWORKS

by

YEHUI ZHANG

A THESIS

Presented to the Department of Computer and Information Science
and the Graduate School of the University of Oregon
in partial fulfillment of the requirements
for the degree of
Master of Science

March 2019

THESIS APPROVAL PAGE

Student: Yehui Zhang

Title: Prediction of ICD-9 Code Assignment Using Attention-based Convolutional Neural Networks

This thesis has been accepted and approved in partial fulfillment of the requirements for the Master of Science degree in the Department of Computer and Information Science by:

Dejing Dou Chairperson

and

Janet Woodruff-Borden Vice Provost and Dean of the Graduate School

Original approval signatures are on file with the University of Oregon Graduate School.

Degree awarded March 2019

THESIS ABSTRACT

Yehui Zhang

Master of Science

Department of Computer and Information Science

March 2019

Title: Prediction of ICD-9 Code Assignment Using Attention-based Convolutional Neural Networks

In intensive care units, most patients are usually in critical conditions which require physicians to make immediate diagnosis and treatments. However, not every patient could get the best treatment because it highly related to the physician's expertise. With the development of the machine learning, many studies have started trying to develop models that can learn the representations in Electronic Health Records (EHR) and make accurate predictions on clinical tasks. On code assignment tasks, models based on convolutional neural networks (CNN) or Recurrent Neural Networks (RNN) have shown promising results but their performances are still insufficient to be applied on real-world applications due to (1) the large number of codes and (2) the length of the document. Here, we propose a Convolutional Neural Network with Multi-label attention mechanism (Multi-Label AT-CNN) model that predict ICD-9 code assignments by learning the base representations of the clinical notes from EHRs.

CURRICULUM VITAE

NAME OF AUTHOR: Yehui Zhang

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

University of Oregon, Eugene
University of Nebraska-Lincoln, Lincoln

DEGREES AWARDED:

Master of Science, Computer and Information Science, 2018, University of Oregon
Bachelor of Science, Biological Systems Engineering, 2015, University of Nebraska-Lincoln

AREAS OF SPECIAL INTEREST:

Deep Learning
Data Mining

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to Professor Dr. Dou for his continuous support and guidance over the past two years.

Last, I wish to thank my family and friends for their never-ending support and encouragement.

TABLE OF CONTENTS

Chapter Page

Chapter Page

LIST OF FIGURES

Figure Page

LIST OF TABLES

Table Page

CHAPTER I

INTRODUCTION

In intensive care units (ICUs), most patients are usually in critical conditions which require physicians to make immediate diagnosis and treatments. However, not every physician can deliver the best treatment to their patients. In fact, there are more than 12 million US adult who were misdiagnosed in outpatient medical care each year (Singh, Meyer, & Thomas, 2014) and more than 40,000 patients die annually due to the misdiagnosis made by physicians (Winters, Custer, & Newman-Toker, 2012). In order to provide fast and accurate diagnosis, people are in hope of relying on computers to solve the problem. With the development of the machine learning, many studies h (Liang, Zhang, Huang, & Hu, 2014) have started trying to develop models that can learn the representations in Electronic Health Records (EHR) and make accurate predictions on clinical tasks. On code assignment tasks, models based on convolutional neural networks (CNN) (Kim, 2014) or Recurrent Neural Networks (RNN) (Choi, Schuetz, Stewart, & Sun) have shown promising results but their performances are still insufficient to be applied on real-world applications due to (1) the large number of codes and (2) the length of the document.

In this study, we propose a Convolutional Neural Network with Multi-label attention mechanism (Multi-Label AT-CNN) model that predict ICD-9 code assignments by learning the base representations of the clinical notes from electronic health records. To evaluate the performance of our model, we use the newest MIMIC-III database(v1.4) which is one of few publicly-available electronic health records databases for conducing scientific researches (Johnson, Pollard, Shen, & Lehman, 2016). All data in MIMIC-III

comes from over 40,000 patients who stayed in critical care units (ICUs) of the Beth Israel Deaconess Medical Center between 2001 and 2012 and those data had been deidentified before publishing. The dataset contains high temporal resolution data including laboratory results, clinical notes, discharge summaries, bedside monitor trends, and waveforms and other biomarkers. Here, we use the discharge summaries to make predictions on the code assignments. The discharge summaries are usually written by physicians at the end of treatment and such documents provides the most accurate and comprehensive information regarding to the actual health condition of patients. From the dataset, we observed that most discharge summaries contain five to twenty ICD-9 codes which makes our tasks as multi-label classification problem. In order to improve the performance on multi-label classification, we adapted per-label attention mechanism which allows our model to learn the distinct representation of each labels.

The structure of the remaining thesis is organized as follows: In Chapter II, we go into details of the background of this study where we list the current challenges we are going to solve, and we briefly introduce the progress of current state-of-art researches that are related to our object. Within those studies, we summarized several recently published papers that are philosophically similar to our work. Chapter III go through the structure of our Attention-Based Convolutional Neural Network Model in details. In Chapter IV, we provide details on the implementation of our model and compare the performance with several baseline models. And the final chapter is the summary of this study with some constructive thoughts on the future works.

CHAPTER II

BACKGROUND & RELATED WORKS

INTERNATIONAL CLASSIFICATION OF DISEASES

Over the past twenty years, electronic health records (EHR) systems, have been widely adapted in hospitals and clinics which routinely collects all health and medicine related data from patients (Adler-Milstein & DesRoches, 2015). Such histories consist of heterogeneous data elements, including patient basic information (age, race, habits, work type), laboratory test results, medicine prescriptions, diagnosis, clinical notes from physicians and nurses, and medical images. Within the system, all types of diseases have been categorized based on the ICD standards (International Statistical Classification of Diseases and Related Health Problems) which is regularized by World Health Organization (International Classification of Diseases, 2018). Although most hospitals in the world had started using the ICD-10 standard, which is more accurate and effective compared the ICD-9, for almost twenty years, most healthcare providers in the United States have less than 10 years history of using ICD-10 standard due to several constrains described in (Johnson G. , 2014). Since most publicly available EHR datasets on deep learning studies use ICD-9, to accurately evaluate the performance with other models, we decide to predict the ICD-9 code assignments based on the available clinical notes.

OVERVIEW OF DEEP LEARNING METHODS

In recent years, deep learning has shown extraordinary performance on many data-related applications, such as machine translation, speech recognition, image classification, and recommendation systems. With its unique structure, deep learning models greatly reduced the efforts on feature engineering and compares to the traditional statistic models or machine learning models, it can learn patterns from very large amount of data in relatively short amount time. In the health care domain, interests in the deep learning methods have rapidly grown because they are capable of generating large complex models based on data that does not require labor-intensive feature engineering from professionals. In recent studies (Purushotham, Meng, Che, & Liu, 2017) (Xiao, Choi, & Sun, 2018) (Huang, Osorio, & Sy, 2018), deep learning models has shown promising performance on mortality prediction, length of stay, and ICD-9 code group production when the non-feature-engineered data were used to train the models. In deep learning, the most fundamental idea is of representation. Before the age of big data, the data used as input features to the machine learning algorithm usually must be hand-engineered from the raw unprocessed data, which heavily relies on the practitioner's expertise and domain knowledges to determine the explicit patterns that meet the interest of the experiment. Such process of creating, analyzing, selecting, or evaluating appropriate features can be extremely labor-intensive and time consuming. Thus, such processes are usually being considered as magic which required creativity with oftentimes luck (Domingos, 2012). Compares to the traditional machine learning methods, deep learning techniques has the ability to learn the optimal features directly from the raw data which does not requires any human guidance. It allows for automatic

discovery the relationship of data that might be difficult or impossible to learn. In this section, we will provide a brief overview of the commonly used deep learning methods on Electronic Health Records system.

Artificial Neural Network

In now days, although new deep learning algorithms and architectures are being proposed continuously, nearly all of them are built upon the artificial neural network (ANN) framework which is composed of interconnected nodes that arranged in layers. In generally, the framework contains three types of layers: the input layer, hidden layers, and output layer. The input layer is start of all the data getting processed and output layer delivers the final prediction. For hidden layers, each layer contains one or mode hidden units which stores a set of weights that can be optimized by minimizing the loss function. The loss functions are usually optimized by the backpropagation algorithm, which is a mechanism for weight optimization that minimizes loss from. This mechanism goes from the final layer of the model backwards through the network. (Goodfellow, Bengio, & Courville, 2016)

From this point, we will introduce several most successful deep learning algorithm and architectures that all developed based on the architecture and optimization procedure of Artificial Neural networks. Those variants been widely used in Electronic Health Record applications to learn the representation of data.

Multilayer Perceptron

Typically, the structure of a multilayer perception (MLP) is just like the ANN which is composed of several hidden layers. Within those layers, every neuron at the same layer is fully connected to neurons of the adjacent layer (Shown in Figure 1). Unlike the recurrent neural networks or undirected deep learning models, due to the limitation of the structure, multiplayer perceptron models usually have only a few hidden layers and data is only capable of flow in one direction. Compares to neuron's updating strategy for artificial neural network, for each hidden unit within the MLP, each neuron in the hidden layer computes a weight sum of the output from the previous layer. Then the sum is applied to a nonlinear activation to generate the output of the neuron. The calculation is represented as the equation shown below:

$$h_i = \sigma(\sum_{n=1}^{N} x_i w_{ij} + b_{ij})$$

Here, N represents the number of units in the previous layer, x_n is the output from the nth unit in the previous layer. w_{ij} and b_{ij} are the corresponding weight and bias that associated with each x_j. Traditionally, to choose the appropriate activation functions, we would use either sigmoid and tanh. As proposed (Goodfellow, Bengio, & Courville, 2016), the rectified linear unit (REL) function becomes more popular in modern MLP models.

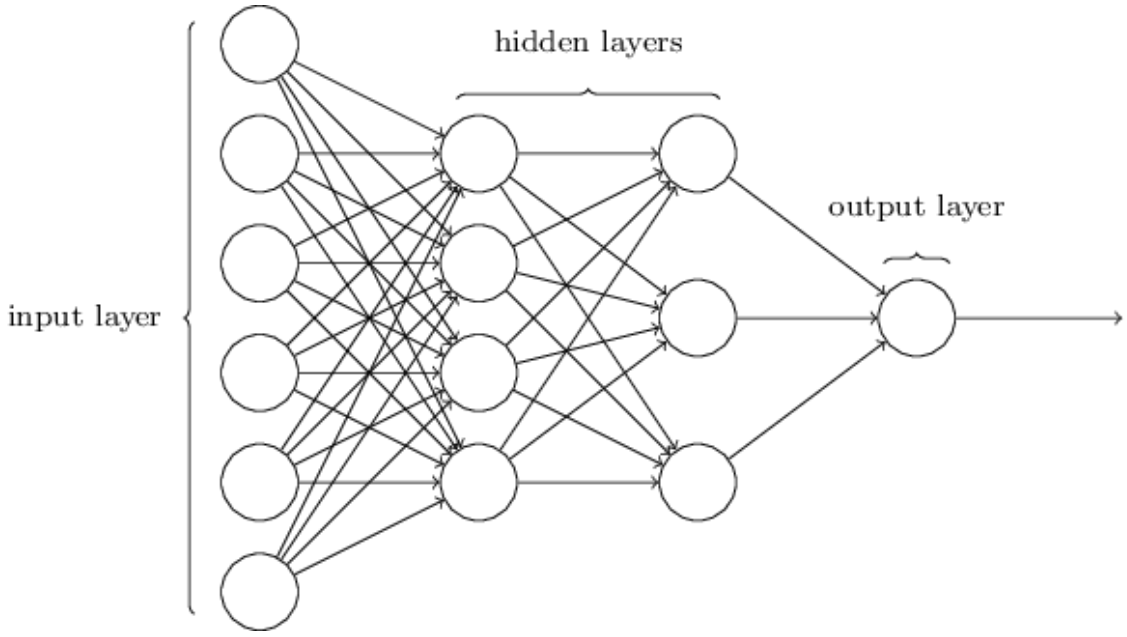

Figure 1 A typical structure of multilayer Perceptron Network where each neuron is fully connected to every neuron of the next layer.

The association between the input data and the predicted values are learned after the weighted are optimized by training the model. As more hidden layers are being added to the model, the model would more likely to learn more complex pattern from the input data which would provide better performance. For multilayer perceptions, the computational cost and training time would also dramatically increase by adding more layers to the model. Hence, such models are generally used to learn relatively simple representation of data.

Convolutional neural networks

With the surging interest of deep learning, convolutional neutral networks (CNNs) have shown extraordinary performance in many domains such as image processing, speech recognition and video analysis. As illustrated in Figure 2, the basic structure of CNN consists of several convolutional layers and those layers are usually followed by its corresponding subsampling layers which is fully connected to the convolutional layer. It often starts with two types of layers: convolutional layers and subsampling layers. The convolutional layers perform convolution operations with several filter maps of equal size. The subsampling layer reduce the sizes of proceeding layers by averaging values within a small neighborhood.

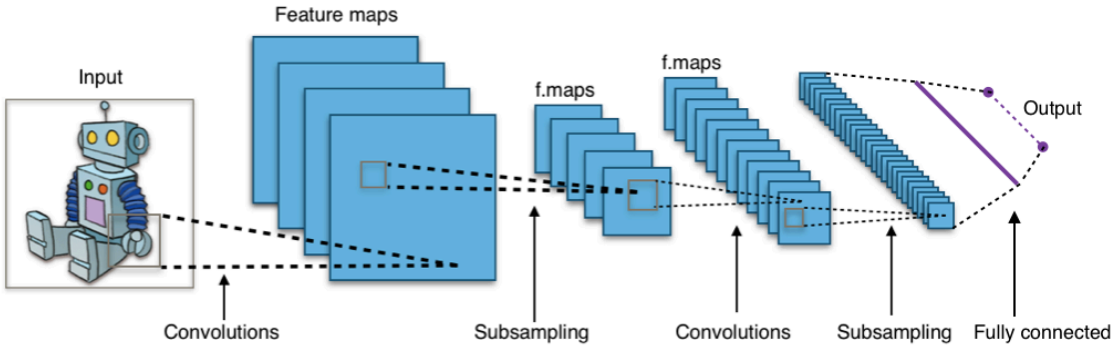

Figure 2 Illustration of typical stature of convolutional neural networks. Source: Wikipedia

The equation for one-dimensional convolution is

$$C_{1d} = \sum_{a=-\infty}^{\infty} x(a)w(t-a)$$

Here, x is the input signal and w is the weighting function which is often called as convolutional filter.

Similarly, the equation for two-dimensional convolution is

$$C_{2d} = \sum_{m} \sum_{n} X(m,n)K(i-m, j-n)$$

Here, X is a two-dimensional grid and K is a filter. At such circumstances, a filter slides a matrix of weight across the entire input to extract the feature maps.

In convolutional neural networks, each unit of the same convolutional layer receives the same input data which comes from the previous layer. Such structure is ideally to extract the lower-level features of the input data from different perspective. After the convolution process, the subsampling layer is applied to aggregate those entreated features. The key parameter to be decided are weights between layers, which are normally trained by standard backpropagation procedures and a gradient descent algorithm with mean squared-error as the loss function. Generally, CNN is designed to learn feature hierarchies without much human interfere which in result provides some degree of translational and distortional invariance.

Recently, one of the most successful CNN models for sentence classification tasks was proposed by (Kim, 2014). It is a simple model with only one layer of convolution that learns the base representation of the document by taking the word vectors trained by *word2vec* method (Mikolov, Chen, Corrado, & Dean, 2013) as the input. This method allows for the use of both pre-trained and task-specific vectors by having multiple channels which greatly accelerates training time and results much better results compares to solely use task-specific vectors. We will use this model as one of the baseline models to compare the performance with our model.

Recurrent neural networks

Recurrent Neural Networks, as one of the most popular architecture being studies, are widely used to model sequential data, such as time series or text. As illustrated in Figure 3, They operate by sequentially updating the hidden state based on the activation of the current input at specific time and the previous hidden state. However, the original implementation of RNN with backpropagation through time algorithm is difficult to learn long-term dependencies due to the vanishing gradient problems. To overcome such problems, two variants of RNN were proposed: Long short-term memory (LSTM) and Gated recurrent unit (GRU). They effectively model sequences of different lengths and capture long range dependencies.

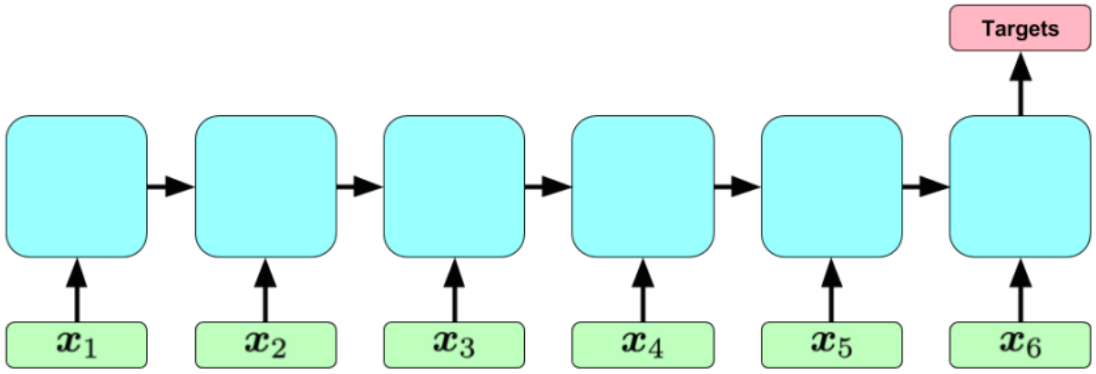

Figure 3 Simple RNN model for multilabel classification. Source: Lipton, 2016

Long Short-Term Memory Networks (LSTM) are one of the variants of Recurrent Neural Networks introduced by Hochreiter and Schmidhuber in 1997 (Hochreiter & Schmidhuber, 1997) which are explicitly designed to avoid the long-term dependency problems. Like the standard RNNs which has a chain of repeating modules structure shown in Figure 1, LSTMs inherited this basic characteristic of Recurrent Neural

Networks, but LSTMs have more complex repeating models than standard RNNs as shown in Figure 2. In this figure, the circles with X-mark inside are element-wise multiplication which are "valves" used to control how much old memory we would like to use from the previous module, how much new memories we generated from the current module should be combined with old memory that would eventually be passed to the next unit, and how much new memory should output to the next LSTM unit. Those implementations are the key ideas of LSTM that overcomes the vanishing gradient problem since we can now restore the shrinking gradient values back to the normal.

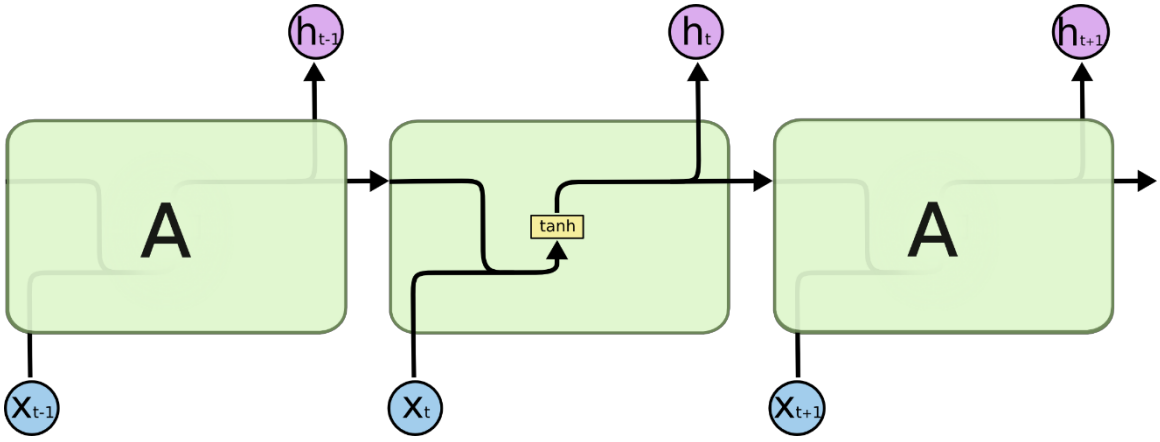

Figure 4 Standard RNN structure contains a single layer, tanh, where X_i are the inputs and h_i are the outputs

Overall, for each LSTM unit, it consists of four different layers:

The first layer is used to generate the vector for the forget which will be used to control how much old memory from previous unit to be passed to the next layer by applying vectors of memory and output from the previous LSTM unit and the input for the current time step to the sigmoid function.

The second layer generates the new memory by applying the output from the previous block and the input of this time step to the tanh as the activation function.

The third layer takes exactly same inputs as the first layer, but the output vector is used as the memory valve which controls how much new memory should be applied on the old memory and the combined memory will eventually be passed to the next LSTM unit. And finally, the last layer is used to generate a vector by applying the same inputs we used in the second layer to the sigmoid as activation function, and then use this vector to control how much newly generated memory from the third layer to be used as output of this block.

Similar to Long Short-Term Memory models, Gated Recurrent Unit (GRU) model, as shown in Figure 5, was designed to adaptively reset or update its memory content. Compares to input and forget gate in LSTM, each GRU has a reset gate and an update date. Differently, GRU fully exposes its memory content at each timestep. By applying the leaky integration, the content from the previous memory unit is balanced with the new memory content.

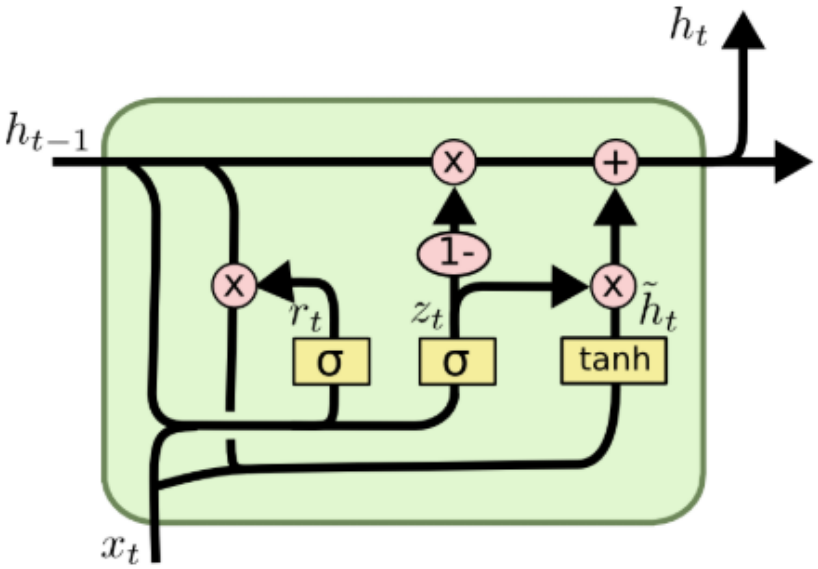

Figure 5 Standard Gated Recurrent Unit Structure which combines the input and forget dates into a single update gate.

DEEP LEARNING APPLICATION ON ELECTRONIC HEALTH RECORDS

In this section, we will provide a thorough review on current state-of-art advances of deep learning methods on Electronic Health record domain.

EHR Information Extraction

The Electronic Health Record (EHR) data is usually consists of structured and unstructured data. The structured data is usually used for billing and administrative purposes such as the admission dates, medication record, and etc. However, most records about the patient's condition are still recorded in the clinical notes which are generally considered as unstructured data. There are several types of clinical notes, such as admission notes, laboratory summaries, discharge summaries, or transfer orders. Since those documents usually are written in free-text style, extracting the useful information from them is usually non-trivial. Before the huge development of deep learning, extracting the information from free-text usually relies heavily on the feature engineering and ontology mappings which makes the cost of such tasks enormously large.

From several recent studies, deep learning methods have been applied on the EHR information extraction domain and the current major tasks are listed in Figure 6.

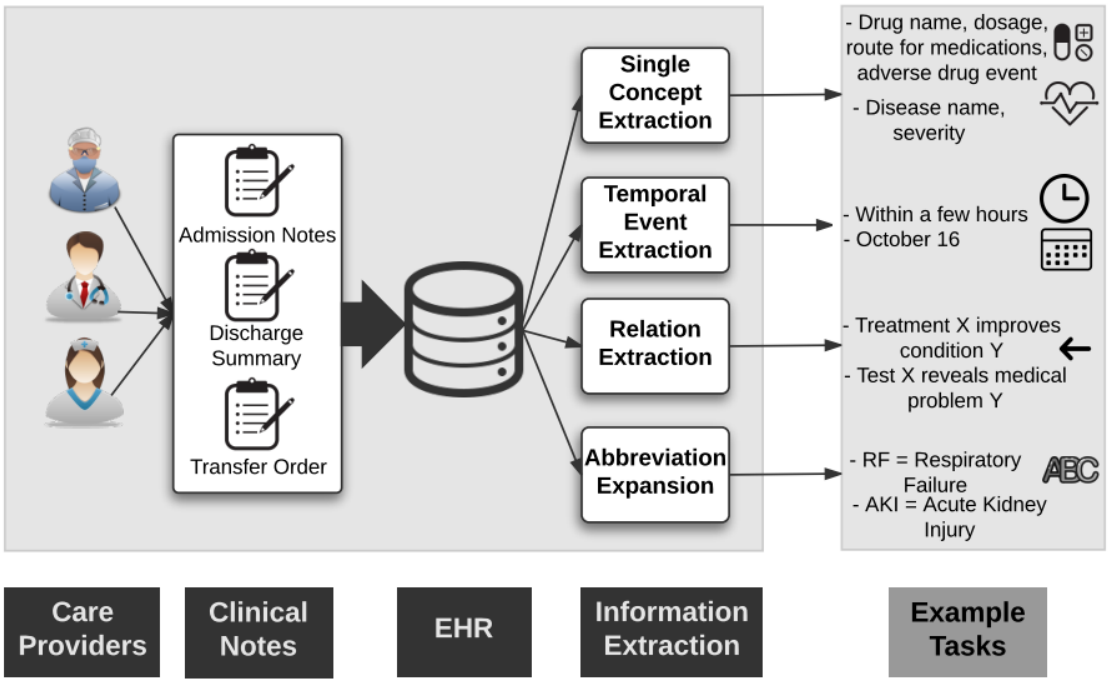

Figure 6 Information Extraction from Electronic Health Record and their relevant tasks.

Single Concept Extraction

Typically, the most valuable information that contained in the clinical notes are related to the disease type, procedure produced, and treatment applied. In recent years, although natural language processing (NLP) techniques had achieved big success on language processing, the results on the information extraction from clinical notes are still not satisfying to be applied in real world applications. In (Jagannatha & Yu, Structured prediction models for RNN based sequence labeling in clinical text., 2016) and (Jagannatha & Yu, Bidirectional RNN for medical event detection in electronic health records., 2016), the single concept extraction can be transformed into the sequence labeling tasks which aims to assign predefined clinical tags to each word exists in the document. Those predefined tags were split into medication and disease categories. For

medication category, they have tags such as prescription name, dosage, usage, and etc. For disease categories, they have tags such as disease name, data of occurrence, severity and etc. In those studies, several Recurrent Neural network variants were development and evaluated which includes traditional LSTM, GRU, bidirectional LSTM. They also tested the model that combines the deep learning models with conditional random fields. In their experiments, their RNN variants models greatly outperformed all of the baseline models and in some specific subdomains, such as extracting the disease severity or duration which related to extracting the numeric values from the free text, their models performed extraordinary performance. A similar research conducted by (Wu, 2015) provides a convolution neural network model which aims to recognize named entity in clinical text also achieved great performance compared to the baselines. Differently, a pre-trained word embedding on clinical text was applied which could be one of the major factors that contributed on the great performance.

Temporal Event Extraction

In general, a temporal event is an event that occurs at the specific time. The extraction of such events is usually more difficult than non-temporal events. In (Fries, 2016), Fries proposed an RNN model which extracts medical events and their corresponding times from the text-text clinical documents. In their study, they utilized a pre-trained word embedding algorithm, word2vec, proposed by (Mikolov, Chen, Corrado, & Dean, 2013), which was already trained with two large clinical corpora. In addition, they used the DeepDive system, developed by Stanford and proposed in (Shin, 2015) for structured relationships and predictions.

Relation Extraction

Similar to the temporal event extraction which extract the association between the clinical events and their corresponding time, the relation extraction tries to find the association between clinical events. In (Lv, Guan, Yang, & Wu, 2016), Lv developed an autoencoder model to learn the data coding in unsupervised manner. In their study, the Unified Medical Language System to utilized to perform the word-concept mapping which allows the autoencoder to generate features. Then the output data was feed in a Conditional Random field classifier.

As discussed in (Liu, Ge, Ji, & McGuinness, 2015), most deep learning architectures proposed on EHR information tasks are evaluated on metrics such as precision, recall, and F1 score. For studies that share similar tasks and evaluation metrics, the reported performance is usually incomparable since they use different dataset on the evaluation step which could product great impact on the performance.

Representation Learning on EHR

As described in previous sections, Electronic Health Record system contains enormous information regarding to medical codes. Those codes were originally used for administrative or billing purposed. Since those codes are closely related to patient's health condition, we can utilize those codes to learn the hidden patterns beneath the data. Inside the EHR system, each medical concept is assigned with a distinct code and those codes usually reflects the relevant ontology of those medical concepts. However, the such relationships represented in hierarchical order failed to represent the similarities between the elements from different coding schemes. Fortunately, due the nature of deep learning models, those hierarchical relationships and representations can be learned by mapping those medical codes into vector space and learned by using deep learning models.

In his section, we will focus on providing an overview of encoding methods on transforming discrete medical codes into vectors with customizable dimensions. Such methods usually rely on unsupervised deep learning architectures which focus on clustering those medical codes based on their natural or designated relationships. Then, we will go over the most recent researches on representing patients using those vectors. Such works usually are supervised deep learning models which aims to make predictions on specific tasks.

Concept Representation

In recent studies such as (Choi, Schuetz, Stewart, & Sun) (Nguyen, Tran, Wickramasinghe, & Venkatesh, 2016), several unsupervised learning methods were developed to derive medical concept vectors by capturing the latent similarities between those concepts. For different medical codes that share similar concepts, their corresponding vectors have relatively close values in lower dimensional vector space. Those vectors then can be analyzed with techniques such as t-Distributed Stochastic Neighbor Embedding, code similarity heatmaps (Mehrabi, Sohn, Li, & Pankratz, 2015) or word-cloud visualization as proposed in (Pham, Tran, Phung, & Venkatesh, 2017).

Distributed Embedding

Within the electronic heath record, a lot of medical events were recorded with time stamps which records the change of the patient's health condition. Such data can form a time series data which can be applied with Natural Language Processing techniques to learn the representation. Generally, for such data structure, NLP techniques can transform the data into fixed-size vector format which often referred as the skip-gram. The skip-gram model was proposed in (Mikolov, Chen, Corrado, & Dean, 2013) as one of the two major implementations for the word embedding framework word2vec. This framework is an unsupervised Artificial Neural Network which transforms large corpus of text into the vector representations. Currently, this framework has been widely used in many deep learning models for text pre-processing and embedding. In (Choi, Bahadori, Searles, Coffey, & Sun, 2016), skip-gram model was used to convert medical codes into distributed code embedding. In the conversion, the skip-gram model heavily

relies on the sequential ordering of the codes. However, in practice, many medical events happen concurrently. To still generate good representation for events happened at the same time, one solution was to group such events as a block. Then for each block, the order of the events is first randomized and then learn the representation for each block. (Choi, Chiu, & Sontag, 2016).

Latent Encoding

Although NLP embedding methods are currently the most popular techniques to learn the representation of the document, other techniques, such as the variant of restricted Boltzmann machine, proposed by. (Tran, Nguyen, Phung, & Venkatesh, 2015), can also delivers promising results. Similarly, from (Lv, Guan, Yang, & Wu, 2016), Lv proposed an autoencoder model which generates the concept vectors from the words extracted from the documents. Their evaluation results demonstrated that autoencoder models outperforms most traditional linear models on representation learning.

Clinical Outcome prediction

Traditionally, there are two major types of approaches for representation learning on electronic health records. One of them is rule-based approach which usually requires medical experts to create the rules based on all the documents (Farkas & Szarvas, 2008). Although this approach is extremely labor intensive and becomes impossible when dealing with massive amount of data, but it is still the most accurate model on ICD-9 assignment tasks. Another type of approach is learning-based approach which usually does not require the model designer have any medical backgrounds. Such approach purely relies on the learning algorithm to find the hidden pattern from the datasets. From one of the previous researches (Lita, Yu, Niculescu, & Bi, 2008), support vector machine (SVM) was used on ICD-9 prediction tasks which shows the potential of learning-based approach on such tasks. One of the major drawbacks of SVM or other machine learning models is that the such models are too "shallow" which is incapable of learning more complex representation from data.

In deep learning studies on Electronic Health Record, the clinical outcome prediction is always one of the most popular subjects which can be divided into two parts. The one part is called static prediction which utilize all the information in the EHR system to make the final prediction. Another part is called temporal outcome prediction where income data keeps feeding into the model, so it has to make predictions continuedly. In general cases, such tasks are handled with unsupervised data modeling techniques, such as the clinical concept representation we presented previous.

Static Outcome Prediction

Compares to the temporal outcome prediction, static outcome prediction is relative easier duo to such tasks does not need to consider the temporal constraints within the data. In (Choi, Schuetz, Stewart, & Sun) , Choi evaluated several models which consists of different Artificial Neural networks with linear models to predict heart failure. They used word embedding method to convert raw clinical documents into word vectors. As the result, they found out that the standard multilayer perceptron outperformed all other ANN variants on heart failure prediction.

From (Tran, Nguyen, Phung, & Venkatesh, 2015), Tran used a different approach on word embedding. They developed a modified Restricted Boltzmann Machine model which takes different set of records from EHR system and output a vector which represents the patient. Then, they used a simple logistic regression classifier to make the prediction on suicide risk prediction task. From their experiment, they reported that the model achieved best performance when all the data related to the patients are used on learning.

In (Miotto, L. Li, Kidd, & Dudley, 2016), Miotto presented a novel unsupervised deep learning method on features that derive a general-purpose patient representation from Electronic Health Record data. They named this method as Deep Patient that facilitates clinical predictive modeling. In this study, the patient vectors are generated with three-layer autoencoder. Then those vectors are used to predict a wide variety of ICD-9 code assignment tasks. Their model showed superior performance on unprocessed clinical features and achieved outstanding result on precision at k metrics.

Similar to the Deep Patient model, in (Liang, Zhang, Huang, & Hu, 2014), Liang used Deep Belief Network to generate patient vectors. The Deep Belief Network is a probabilistic generative model that are composed of several layers of stochastic latent variables. Similar to autoencoders or Restricted Boltzmann machine, each hidden layer of the sub-network within the DBN is visible to the next layer. The generated vectors are feed into a support vector machine for disease code prediction. Similarly, (Li, Li, Ramanathan, & Zhang, 2014) used a two-layer DBN for identifying osteoporosis. The framework used a discriminative learning approach where top risk factors were identified based on DBN reconstruction errors. By utilizing all the identified risk factors, the model resulted the best performance over all other baseline models.

Compare to the studies above that utilizes heterogeneous data from the EHR system, some studies used clinical note solely on the clinical outcome prediction tasks. In (Jacobson & Dalianis, 2017), they proposed a stacked Restricted Boltzmann Machine model and stacked Autoencoder with word2vec word embedding approach to make predictions on the healthcare associated infections. Their result showed that the stacked RBM model yield the best F1 score with raw clinical and the stacked autoencoder model achieved best performance when the documents are embedded with word2vec.

Temporal Outcome prediction

Empirically, the temporal outcome prediction tasks can be categories into two types. One type is to predict the outcome within a certain time interval which is similar to the static outcome prediction. Another type is to make prediction based on time series data.

In recent studies, both LSTM and GRU have shown promising performance on code assignment tasks. In (Lipton, Kale, Elkan, & Wetzell, 2016), authors used LSTM to classify 128 diagnoses based on 13 frequently but irregularly sampled clinical measurements. Those measurements are sampled time series data which includes body temperature, heart rate, diastolic and systolic blood pressure, and blood glucose. In their study, they achieved best performance by applying ensemble method on the combination of Long Sort Term Memory with a standard three-layer multilayer perceptron. In (Cheng, Wang, Zhang, Xu, & Hu, 2015), Cheng trained a convolutional neural network on temporal matrices of clinical codes. Its model aims to predict the onset of both congestive heart failure and chronic obstructive pulmonary disease.

In (Choi, Bahadori, Searles, Coffey, & Sun, 2016), they developed a system called Doctor AI which is a generic predictive model that aims to perform multilabel prediction over time while sequence labeling task predicts a single label at each step. A Gated Recurrent Unit network was trained which aims to predict the next coded event by taking the previous observed time sequence. As they claimed, their system could produce similar accuracy compared to human physicians. They also tested their model on publicly available dataset MIMIC-III and outperformed all other similar models on the same dataset. On MIMIC-III dataset, they first pre-train their model with their private dataset,

then evaluate their model. In addition, they also trained a Gated Recurrent Unit network on the sequences of clinical event vectors that was derived from word2vec skip-gram implementation. They claimed that their model achieved superior performance over other baselines models for predicting the onset of heart disease during various prediction windows.

In (Pham, Tran, Phung, & Venkatesh, 2017), Pham proposed a Deep Care framework which derives clinical concept vectors by using the word2vec framework to embed the clinical document. Differently, they created two separate vectors for each patient admission. The first vector is used for diagnosis codes and the second one is for intervention codes. Then they concatenated those vectors into one and pass into an LSTM network for predicting the next diagnosis and next intervention for both diabetes and mental health cohorts. They model disease progression by examining precision @ k metrics for all prediction tasks. They also predict future readmission based on these past diagnoses and interventions. For all tasks, they found the deep approaches resulted in the best performance.

In (Nguyen, Tran, Wickramasinghe, & Venkatesh, 2016), they developed a system that uses the Convolutional neural networks for predicting the unplanned re-admission after the patients was being discharged from the hospital. Like other outcome prediction models, it utilizes the discrete clinical event codes as the input data. They claimed that their model is superior to the bag-of-codes model and other traditional machine learning baseline models because their model could still achieve good performance when the gap between the nearest two clinical events are far part. Similarly, in (Esteban, Staeck, Yang, & Tresp, 2016), they proposed several deep learning models

for predicting the onset of complications relating to kidney transplantation. From the EHR system, they derived both static and dynamic features into vectors and used them as input for various variants of RNN architecture. After the experiments, they found that the Gated Recurrent Unit network works best than other RNN variants and baseline models when only static features were used as the input data. They claimed that if long term dependencies were not considered as important relations, using embedded static features results in improved performance. Otherwise, the dynamic embedded data could perform better if time dependencies matter.

In (Che, Purushotham, Cho, Sontag, & Liu, 2018), they developed a variation of the recurrent gated unit model which tried to overcome the performance drop when there are missing values on the clinical time series data. The model takes two representations of the miss pattern: masking and time interval that captures the long-term temporal dependencies in time series. Their evaluation showed that their model improved AUC on two real-word ICD-9 classification and mortality prediction tasks.

While there are numerous ways of making clinical outcome predictions, most methods that involved deep learning architectures are evaluated with standard classification metrics such as AUC, accuracy, and precision, recall, and F1 score. For temporal prediction tasks, precision and recall are the two major matrices are being used.

In this study, we are especially interested in predicting the ICD-9 codes based on the clinical notes from Electronic Health Records databases. It is the ninth revision of International Statistical Classification of Diseases and Related Health Problems which was widely used in U.S. hospitals before 2014. In the EHRs, clinical notes are usually written by physicians or nurses in free-text format. Such format indicates that the data in the notes are not structured. Since almost all the EHR systems contains private information about patients and de-identifying those records are extremely labor intensive, they are only a few publicly available datasets can be used. MIMIC-III, Medical Information Mart for Intensive Care 3rd revision, is one of the best publicly available datasets which contains complete EHRs from more than 4,000 patients who stayed in critical care units of the Beth Israel Deaconess Medical Center between 2001 and 2012 (Johnson, Pollard, Shen, & Lehman, 2016).

On multi-label prediction tasks, the major drawback of the traditional CNN approaches is that they simply encode all the words appears in the document and feed them into the model all at once regardless of the sequence of the data. Since the sequence of the data is also critical on such tasks, using RNN models would be considerable. However, one potential issue of training the RNN models with the clinical notes is that since each document consists of thousands of words, if each unit of the RNN only takes one word as the input, the training time of the model would be unacceptable. To overcome this problem, a hierarchical Attention-bidirectional Gated Recurrent Unit model (HA-GRU) was recently proposed by (Baumel, Nassour-Kassis, Cohen, & Elhadad, 2017) which uses two Gated Recurrent Unit that encodes both words and

sentences which greatly reduces length of the sequence to the model so it can be trained much faster and still produces promising results.

As shown in Figure 7, initially, each word in the document was embedded by using Continuous Bag-of-Words method (Mikolov, Chen, Corrado, & Dean, 2013). Then the first GRU is used to encode those embedded words into vectors where each vector represents one sentence. The vector is then encoded using a neural attention mechanism. Since problem is categorized as multi-label classification, each sentence vector would be feed into the different attention layers to generate class-specified encoding. Then those outputs are applied to a fully connected layer with softmax for each classifier to determine the label. In this model, attention mechanism is applied to both sentence encoder and classifier. In (Shen et al 2014. Gao et al 2014), authors demonstrated that deep learning models with attention mechanisms usually results in better performance since it provides the insight of which elements serves more important than others on the prediction. In Baumel's study, the attention model in the classifier was used to track the sentences with the attention score for each label. Sentences with higher attention score usually contributes the most on the decision making of the label prediction. Similarly, the attention model in sentence encoder can be used to track the word that contributed most to the final prediction. With both attention models, the final prediction becomes explainable which greatly helps medical experts to better understand the causes of the symptoms.

In summary, on multi-label prediction tasks such as predicting the ICD-9 codes based on the clinical notes, most current state-of-art models can either (1) achieve high accuracy on the but lack of explainable result and sometimes requires huge amount of

computational resources for training, such as (Kim, 2014) and (Choi, Schuetz, Stewart, & Sun) (2) provide explainable result on the prediction but lacks on the performance, such as (Song, Rajan, & Spanias, 2018) (2) provide explainable result but takes too long for training, such as (Baumel, Nassour-Kassis, Cohen, & Elhadad, 2017). Compares to the "black box" models that does not provide explainable result, we believe the explainable deep learning models on ICD-9 code prediction tasks can be much beneficial to medical community to better understand the causes of the disease. Thus, on top of developing a model can provides explainable results, we want to develop a model that achieve better performance and less processing time compares to the current state-of-art models.

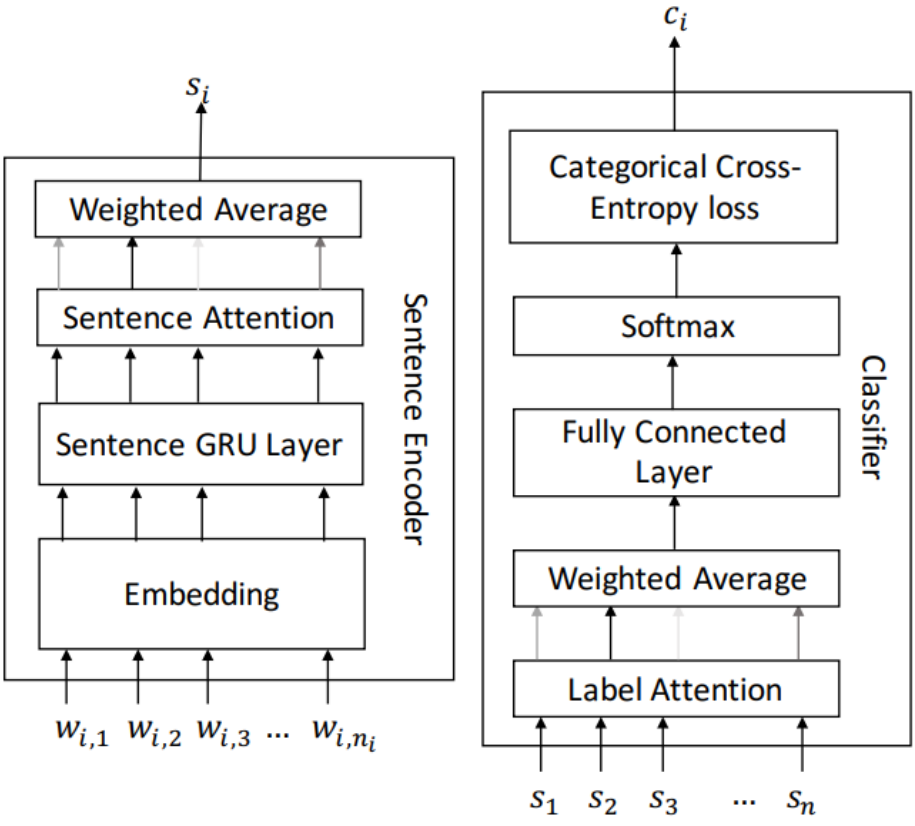

Figure 7 Hierarchical Attention-Bidirectional Gated Recurrent Unit model for code assignment tasks. Source: Baumel, 2017

CHAPTER III

METHODOLOGY

In this section, we will go into details on the structure of our model. First, the input data are discharge summaries from EHRs which are in free-text format. After preprocessing the documents into arrays of words, we perform word embedding for each word and get $X \in \mathbb{R}^{k*N}$ where k is the dimension of the word embedding and N is the length of the document. The details on word preprocessing and embedding are described in Chapter IV. For each input entry, it has the output data $Y \in \mathbb{R}^{c}$ where c is the number of labels we will predict.

CONVOLUTIONAL NEURAL NETWORK

To learn the base representation in sentence level, we concatenate adjacent words from the document together to form

$$x_{i:i+f-1} = x_i \oplus x_{i+1} \oplus \dots \oplus x_{i+f-1}$$

where f is the width of the filter and $x_{i:i+f-1}$ represents the concatenation of f words start at position i. Then, we used the method proposed by (Kim, 2014) to embed adjacent word vectors by passing the input data through the convolutional neural network illustrated in where the filter

$$w \in \mathbb{R}^{k*f}$$

At each step n, we compute

$$h_n = f(w * x_{i:i+f-1} + b)$$

where $b \in \mathbb{R}$, f is a non-linear function such as the hyperbolic tangent. In order to get the output $H \in \mathbb{R}^{f*N}$ where $h_n \in H$ and H, we pad zeros at the end of the vector for all the $x_{i:i+f-1}$ where $i + f - 1 < N$ so there would have exact N vectors being embedded by the CNN.

After the convolution, we want to apply a per-label attention mechanism which learns different part of the base representation based on the label it predicts. With such mechanism, we can also get the adjacent word vectors that are most relevant to the label we are trying to predict. Hence, for each label c, we want to get the distribution of the location for each label in the document by applying the softmax operation to the matrix-vector product of the representation we get by convolution with a vector parameter for label c. To formalize, to compute the attention vector D_c for label c,

$$D_c = Softmax(H^T \mathcal{E}_c)$$

where $\mathcal{E}_c \in \mathbb{R}^f$ is the vector parameter for c and $H^T \in \mathbb{R}^{N*f}$ is the transpose of the embedded adjacent words vector and $D_c \in \mathbb{R}^N$.

Next, for each label, we can get the vector representation of the entire document by summing the product of the attention vector with each adjacent word vectors.

$$r_c = \sum_{n=1}^{N} d_{c,n} h_n$$

where $d_{c,n} \in D_c$ and r_c is the vector representation of label c in the document.
Finally, we can compute the probability for each label c by applying a linear layer and a sigmoid transformation to the r_c:

$$\hat{y}_c = \sigma(\alpha_c^T r_c + b_c)$$

where α_c^T is a vector of prediction weights and b_c is the bias.

To train the model, we are trying to minimize the binary cross-entropy loss with L2 norm using Adam optimizer. (Kingma & Ba, 2015)

$$Loss(X, y) = -\sum_{c=1}^{C} y_c \log(\hat{y}_c) + (1 - y_c)\log(1 - \hat{y}_c)$$

where $\hat{y}_c$ is the predicted probability label c and $y_c \in \{0,1\}$ is the actual value of label c.

The overall model is illustrated in Figure 9 and Figure 8.

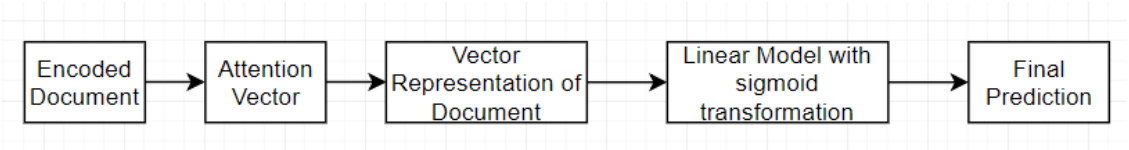

Figure 8 Multi-Label AT-CNN: Part 2. For each label, we apply its corresponded attention vector to the encoded document to get the overall representation of document for the label. Then the linear model with sigmoid transformation are applied to make the final result in range 0 to 1.

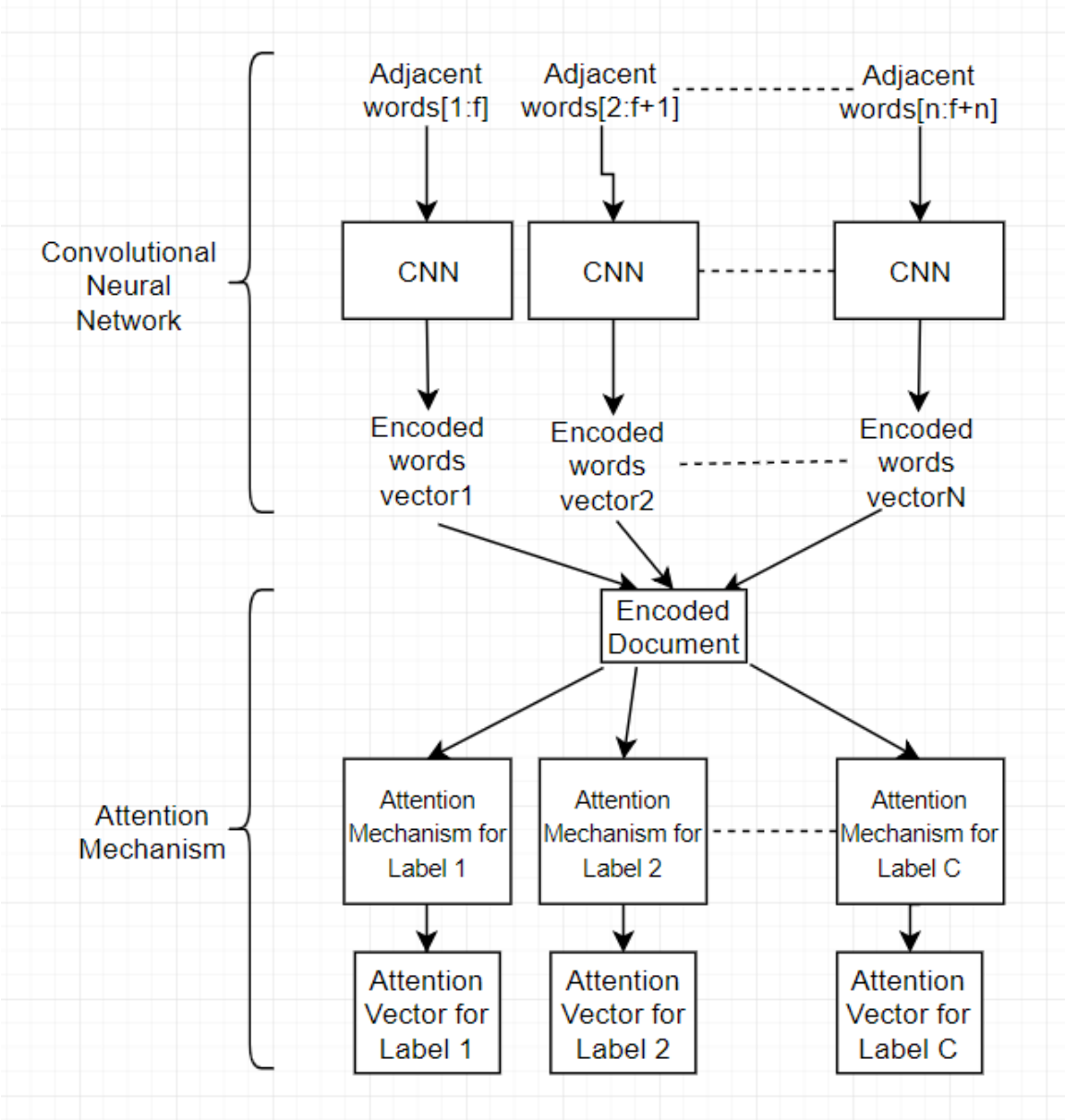

Figure 9 Multi-Label AT-CNN: Part 1. Using CNN to encode the document and Attention Mechanism to learn the relevance of each adjacent word vectors to each label.

CHAPTER IV

EVALUATION

DATA PREPROCESSING

Since all the discharge summaries from the MIMIC-III database are in free-text format and some components in the document would negatively impact the performance of the training model, we want to remove all the unwanted components from the documents before training. First, we split the entire documents by space into an array and we consider each element of the array as a token. For each token, we converted all the capital letters to lowercase, remove all the punctuation symbols and for simplicity and efficiency, we removed all the tokens that contains numeric values. Then we count the appearance of each word in the entire dataset and converted words that appeared less than 5 times to the specifically designated token "UNK" which stands for unknown. After preprocessing each document, we perform word embedding using word2vec framework which converts the word into a k-dimensional vector. By applying word embedding, the vectors with similar meanings would have smaller difference than the difference with other words.

As described in the (Mikolov, Chen, Corrado, & Dean, 2013), word2vec model has two implementations: Continuous Bag-of-Words (CBOW) and skip-gram. The major difference between the CBOW and skip-gram is that CBOW uses the word sequences before and after the target word to make the prediction and skip-gram uses one word to predict the preceding word sequences. As compared in (Xiao, Choi, & Sun, 2018), skip-gram usually serves better performance to represent infrequent words. However,

compares to CBOW, it takes longer time for training. In our study, we separately evaluate both implementations and their performances are reported in Appendix C.

After the tokens are being embedded, we get the output $X = [x_1, x_2, \dots, x_N]$ where N is the number of tokens in each document and $x_i \in \mathbb{R}^k$ where k is the dimension of the word vector. We tried k with different values which includes 100, 200, 400, and 800 and as the result, when k = 400, our model works best.

For the label we are trying to predict, we extracted all the ICD-9 codes from the discharge summaries. For each document, we created a set which contains all the ICD-9 code appeared in the document. Furthermore, we created another set associated with the document which contains the categories of the codes from the set we created. Then, we sum up all the codes and categories, separately and the statistics is shown in Table 1. From the table, the Top-10 Codes dataset represents the top 10 most frequent diseases appeared in the overall discharge summaries. The names of top-10 codes and categories can be found at Appendix A Table 4. By comparing the most frequent codes with the code set we got for each documents and the most frequent categories with the category set, we noticed that those codes and categories shown in Table 4 are usually just the symptoms of the disease which appears much less frequently than others. Therefore, based on the experience from past and the report from (Huang, Osorio, & Sy, 2018), we know that although predicting the top-10 codes or categories would result much better result than predicting top-50s, the result from predicting the top10s has much less impact and meaning for real world applications. Hence, in this study, we would only predict the top-50 codes and categories and we built separately models for each dataset.

Datasets	Number of Admission	Coverage
Top-10 Codes	40562	76.93%
Top-10 Categories	44419	84.24%
Top-50 Codes	49354	93.60%
Top-50 Categories	51034	96.76%

Table 1 The statistics of the datasets from MIMIC-III discharge summaries

To prepare the training and testing dataset, we used 10-fold cross-validation method from Scikit-Learn (Pedregosa, 2011) which is a publicly available python library for machine learning.

PARAMETER TUNING

Initially, we have the following hyperparameters setting: dropout rate = 0.5; learning rate = 0.001; filter size: 5; number of filters: 100. To fine tuning the parameters and rates of our model, we used the Spearmint which is a python package to perform Bayesian optimization according to the algorithms outlines in (Larochelle & Adams, 2012). The final hyperparameters that delivers the best performance are as follow: dropout rate: 0.3; learning rate: 0.0001; filter size: 10; number of filters: 75;

EVALUATION METRICS

In this study, we used the same evaluation metrics that had been used in (Huang, Osorio, & Sy, 2018) (Purushotham, Meng, Che, & Liu, 2017) which includes Micro-averaged and Macro-Averaged Area Under the Receiver operating characteristic curve (AUROC) score and F1 score. The macro-averaged AUROC computes the metric

independently for each object and then take the average of all objects. Differently, the micro-averaged AUROC compute the average metric after aggregate the contributions of all classes. The formula to compute Micro-AUROC and Macro-AUROC are as follow:

$$MicroAUROC = \frac{\sum_{c=1}^{C} P_c}{\sum_{c=1}^{C} P_c + N_c}$$

$$MacroAUROC = \frac{1}{C}\sum_{c=1}^{C} \frac{P_c}{P_c + N_c}$$

where P_c is the count of positive examples for label c and N_c is the count of negative examples.

RESULTS

The baseline models we used are Logistic Regression, Convolutional Neural Network and a hierarchical Attention-bidirectional Gated Recurrent Unit model (HA-GRU). For the Logistic regression, we converted discharge summaries to embedded unigram vectors as input. Then we used the One-vs-rest logistic regression model implemented in (Pedregosa, 2011) to make predictions. The CNN model was implemented based on the (Kim, 2014) and the structure of the model that results the best performance is shown in Appendix A. Last, the RNN model, HA-GRU, which is currently one of the best models on ICD-9 code prediction tasks, was implemented based on (Baumel, Nassour-Kassis, Cohen, & Elhadad, 2017).

In Table 2 and Table 3, we list the best performance that our models could achieve. Although for some evaluation metrics, RNN models showed slightly better

performance. The model we proposed generally only uses 60% training time compares to most RNN models. The complete results of models with different size of the embedding dimensions are listed in Appendix B. From those tables, we observed that 400 dimensions word embedding achieved best result on both CNN and RNN models. Moreover, our CNN models showed slightly better performance on predicting Top-50 code than predicting Top-50 categories, but the RNN models showed better performance on predicting Top-50 categories.

Models	F1	AUROC Macro	AUROC Micro
Logistic Regression	0.4026	0.65	0.918
CNN	0.4224	0.778	0.924
HA-GRU	0.4774	0.806	**0.954**
Multi-Label AT-CNN	**0.4862**	**0.814**	0.948

Table 2 F1 and AUROC scores of ICD-9 Top-50 Code Prediction tasks on test set from MIMIC-III Clinical Notes

Model	F1	AUROC Macro	AUROC Micro
Logistic Regression	0.417	0.674	0.927
CNN	0.43	0.794	0.942
HA-GRU	**0.491**	0.822	**0.963**
Multi-Label AT-CNN	0.485	**0.837**	0.962

Table 3 F1 and AUROC scores of ICD-9 Top-50 category Prediction tasks on test set from MIMIC-III Clinical Notes

CHAPTER V

CONCLUSION

In this study, we proposed a learning-based automatic ICD-9 code assignment model that outperforms most rule-based and learning-based models on code assignment task. Our model consists a simple convolutional neural network with attention mechanism that performs multi-label prediction based on free-text documents. Our model yields strong improvements over previous metrics on several formulations of the ICD-9 code prediction tasks, while providing satisfactory explanations for its prediction. Although we focus on the prediction of ICD-9 codes, our model can also made ICD-10 code predictions without modification or perform real-time diagnosis recommendation based on the clinical notes. Compares to the recurrent neural network models, one of the advantages of our model is that it can also learn the base representation of document on sentence level, but our CNN model structure is much simpler than RNN models which greatly reduces the training time.

During processing our datasets, we had noticed that the diagnosis codes in the documents are also have hierarchies. For example, a lot of ICD-9 code are medical condition, like hypertension. Sometimes, a patient could be assigned to numerous codes but sometimes only one code is the cause of all other codes. For examples, thrombus, also known as blood clot, is the final product of blood coagulation step in hemostasis. It could cause trauma, hypertension. Since those side effects are commonly shared with other diseases and are frequently appears in most ICU patients, if our model could only predict the symptoms without correctly predicting the causes, then our model could have much less usefulness for real world applications. In current studies, all the diseases codes are

treated equally. Therefore, in the future work, we should build a model that could assign weights to the labels based on the hierarchy of the codes.

Moreover, most studies on the ICD-9 code prediction tasks uses discharge summaries which is usually written after the patient was finished the session. Although discharge summaries provide the information that is nearest to the ground truth of patient's condition, we should also try to build the model that takes time series dataset to make the prediction, so it could continuously provide recommendation of diagnose to medical stuffs.

APPENDIX A

Top-10 Codes	Coverage	Top-10 Categories	Coverage
Hypertension	38.01%	Essential hypertension	39.15%
Congestive heart failure	24.35%	Cardiac dysrhythmias	31.81%
Atrial fibrillation	23.87%	Disorders of fluid electrolyte	27.90%
Coronary atherosclerosis	23.09%	Disorders of lipoid metabolism	26.95%
Acute kidney failure	16.89%	Other chronic ischemic heart disease	26.70%
Diabetes Type II	16.65%	Diabetes mellitus	26.20%
Hyperlipidemia	16.12%	Heart failure	25.28%
Acute respiratory failure	13.75%	Other diseases of lung	24.65%
Urinary tract infection	12.22%	Other and unspecified anemias	23.52%
Esophageal reflux	11.67%	Acute kidney failure	21.14%

Table 4 The coverage of most frequent disease code and categories in the MIMIC-III discharge summary

APPENDIX B

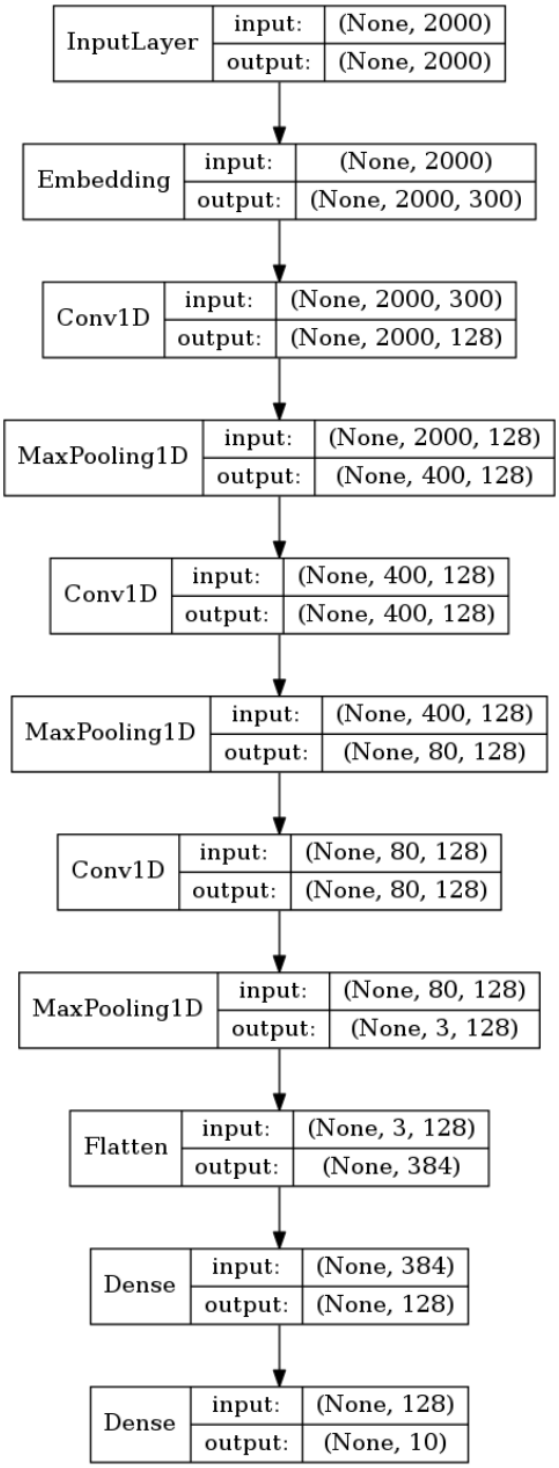

Figure 10 Convolution Neural Network Architecture for ICD-9 code prediction

APPENDIX C

Model	Embedding method	F1	AUROC Macro	AUROC Micro
Logistic Regression	CBOW(k=100)	0.368	0.543	0.853
	CBOW(k=200)	0.419	0.626	0.767
	CBOW(k=400)	0.399	0.597	0.948
	CBOW(k=800)	0.379	0.668	0.943
CNN	CBOW(k=100)	0.424	0.716	0.902
	CBOW(k=200)	0.422	0.778	0.924
	CBOW(k=400)	0.415	0.768	0.913
	CBOW(k=800)	0.425	0.754	0.915
HA-GRU	CBOW(k=100)	0.377	0.757	0.914
	CBOW(k=200)	0.421	0.727	0.868
	CBOW(k=400)	0.454	0.792	0.906
	CBOW(k=800)	0.425	0.716	0.902
Multi-Label AT-CNN	CBOW(k=100)	0.380	0.715	0.904
	CBOW(k=200)	0.422	0.752	0.921
	CBOW(k=400)	0.486	0.814	0.948
	CBOW(k=800)	0.475	0.811	0.934

Table 5 Performance of models on Top-50 Code Prediction with input embedded with Continuous Bag-of-Words method

Model	Embedding method	F1	AUROC Macro	AUROC Micro
Logistic Regression	SG(k=100)	0.390	0.571	0.856
	SG(k=200)	0.435	0.604	0.875
	SG(k=400)	0.425	0.634	0.905
	SG(k=800)	0.403	0.650	0.918
CNN	SG(k=100)	0.403	0.736	0.859
	SG(k=200)	0.425	0.708	0.847
	SG(k=400)	0.394	0.739	0.867
	SG(k=800)	0.404	0.790	0.873
HA-GRU	SG(k=100)	0.435	0.741	0.914
	SG(k=200)	0.474	0.765	0.932
	SG(k=400)	0.477	0.806	0.954
	SG(k=800)	0.462	0.804	0.950
Multi-Label AT-CNN	SG(k=100)	0.355	0.691	0.859
	SG(k=200)	0.365	0.786	0.875
	SG(k=400)	0.462	0.841	0.815
	SG(k=800)	0.414	0.770	0.887

Table 6 Performance of different models on Top-50 Code Prediction with input embedded with Skip-Gram method

Model	Embedding method	F1	AUROC Macro	AUROC Micro
Logistic Regression	CBOW(k=100)	0.381	0.550	0.771
	CBOW(k=200)	0.397	0.622	0.796
	CBOW(k=400)	0.407	0.630	0.814
	CBOW(k=800)	0.417	0.674	0.927
CNN	CBOW(k=100)	0.423	0.718	0.920
	CBOW(k=200)	0.430	0.794	0.942
	CBOW(k=400)	0.425	0.783	0.931
	CBOW(k=800)	0.426	0.779	0.930
HA-GRU	CBOW(k=100)	0.386	0.776	0.945
	CBOW(k=200)	0.429	0.741	0.885
	CBOW(k=400)	0.471	0.817	0.924
	CBOW(k=800)	0.442	0.725	0.920
Multi-Label AT-CNN	CBOW(k=100)	0.388	0.729	0.922
	CBOW(k=200)	0.430	0.767	0.939
	CBOW(k=400)	0.486	0.830	0.953
	CBOW(k=800)	0.485	0.837	0.962

Table 7 Performance of models on Top-50 Category Prediction with input embedded with Continuous Bag-of-Words method

Model	Embedding method	F1	AUROC Macro	AUROC Micro
Logistic Regression	SG(k=100)	0.397	0.588	0.873
	SG(k=200)	0.404	0.606	0.893
	SG(k=400)	0.414	0.647	0.904
	SG(k=800)	0.425	0.675	0.916
CNN	SG(k=100)	0.411	0.721	0.863
	SG(k=200)	0.424	0.722	0.864
	SG(k=400)	0.405	0.747	0.871
	SG(k=800)	0.409	0.736	0.874
HA-GRU	SG(k=100)	0.444	0.750	0.923
	SG(k=200)	0.471	0.793	0.946
	SG(k=400)	0.491	0.822	0.963
	SG(k=800)	0.482	0.823	0.962
Multi-Label AT-CNN	SG(k=100)	0.361	0.705	0.866
	SG(k=200)	0.386	0.807	0.877
	SG(k=400)	0.471	0.845	0.832
	SG(k=800)	0.431	0.813	0.920

Table 8 Performance of models on Top-50 Category Prediction with input embedded with Skip-gram method

REFERENCES CITED

Adler-Milstein, J., & DesRoches, C. (2015). Electronic Health Record Adoption In US Hospitals: Progress Continues, But Challenges Persist. *HEALTH AFFAIRS*.

Baumel, T., Nassour-Kassis, J., Cohen, R., & Elhadad, N. (2017). Multi-Label Classification of Patient Noptes: Case Study on ICD Code Assignment. *arXiv:1709.09587v3*.

Che, Z., Purushotham, S., Cho, K., Sontag, D., & Liu, Y. (2018). Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*.

Cheng, Y., Wang, F., Zhang, P., Xu, H., & Hu, J. (2015). Risk Prediction with Electronic Health Records : A Deep Learning Approach. *International Conference on Data Mining*.

Choi, E., Bahadori, M. T., Searles, E., Coffey, C., & Sun, J. (2016). Multi-layer Representation Learning for Medical Concepts. *arXiv:1602.05568*.

Choi, E., Schuetz, A., Stewart, W. F., & Sun, J. (n.d.). Medical Concept Representation Learning from Electronic Health Records and its Application on Heart Failure Prediction. *arXiv:1602.03686*.

Choi, Y., Chiu, C. Y.-I., & Sontag, D. (2016). Learning Low-Dimensional Representations of Medical Concepts Methods Background. *AMIA Summit on Clinical Research Informatics*.

Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*.

Esteban, C., Staeck, O., Yang, Y., & Tresp, V. (2016). Predicting Clinical Events by Combining Static and Dynamic Information Using Recurrent Neural Networks. *arXiv:1602.02685*.

Farkas, R., & Szarvas, G. (2008). Automatic construction of rule-based ICD-9-CM coding systems. *BMC bioinformatics*.

Fries, J. A. (2016). Brundlefly at SemEval-2016 Task 12: Recurrent neural networks vs. joint inference for clinical temporal information extraction. *n Proceedings of the 10th International Workshop on Semantic*.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. *MIT Press*.

Hazlewood, A., & FAHIMA, R. (2003). ICD-9 CM to ICD-10 CM: Implementation Issues and Challenges. *AHIMA's 75th Anniversary National Convention and Exhibit Proceedings*.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation, 9*(8), 1735-1780.

Huang, J., Osorio, C., & Sy, L. W. (2018). An Empirical Evaluation of Deep Learning for ICD-9 Code Assignment using MIMIC-III Clinical Notes. *arXiv:1802.02311*.

International Classification of Diseases. (2018, 12 6). Retrieved from World Health Organization: https://www.who.int/classifications/icd/en/

Jacobson, O., & Dalianis, H. (2017). Applying deep learning on electronic health records in Swedish to predict healthcare-associated infections. *Association for Computational Linguistics*.

Jagannatha, A. N., & Yu, a. H. (2016). Bidirectional RNN for medical event detection in electronic health records. *Association for Computational Linguistics*.

Jagannatha, A. N., & Yu, H. (2016). Structured prediction models for RNN based sequence labeling in clinical text. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Johnson, A., Pollard, T., Shen, L., & Lehman, L. (2016). MIMIC-III, a freely accessible critical care database. *Scientific Data*.

Johnson, G. (2014). The Impact of ICD-10 Implementation on Hospital Providers. *Applied Research Projects*.

Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 1746-1751.

Kingma, D., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations*.

Larochelle, J. S., & Adams, R. (2012). Practical Bayesian Optimization of Machine Learning Algorithms. *Advances in Neural Information Processing Systems*.

Li, H., Li, X., Ramanathan, M., & Zhang, A. (2014). Identifying informative risk factors and predicting bone disease progression via deep belief networks. *Methods*.

Liang, Z., Zhang, G., Huang, J. X., & Hu, Q. V. (2014). Deep learning for healthcare decision making with EMRs. *IEEE International Conference on Bioinformatics and Biomedicine*.

Lipton, Z. C., Kale, D. C., Elkan, C., & Wetzell, R. (2016). Learning to diagnose with LSTM recurrent neural networks. *International Conference on Learning Representations*.

Lita, L. V., Yu, S., Niculescu, S., & Bi, J. (2008). Large scale diagnostic code classification for medical patient records. *Proceedings of the Third International Joint Conference on Natural Language Processing*.

Liu, Y., Ge, T., Ji, H., & McGuinness, D. (2015). Exploiting Task-Oriented Resources to Learn Word Embeddings for Clinical Abbreviation Expansion. *Proceedings of the 2015 Workshop on Biomedical Natural Language Processing*.

Lv, X., Guan, Y., Yang, J., & Wu, J. (2016). Clinical Relation Extraction with Deep Learning. *International Journal of Hybrid Information Technology*.

Mehrabi, S., Sohn, S., Li, D., & Pankratz, J. J. (2015). Temporal Pattern and Association Discovery of Diagnosis Codes Using Deep Learning. *Proceedings of 2015 International Conference on Healthcare Informatics*.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *arXiv:1301.3781*.

Miotto, R., L. Li, Kidd, B. A., & Dudley, J. T. (2016). Deep Patient: An Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records. *Scientific reports*.

Nguyen, P., Tran, T., Wickramasinghe, N., & Venkatesh, S. (2016). Deepr: A Convolutional Net for Medical Records. *arXiv:1607.07519*.

Nickerson, P., Tighe, P., Shickel, B., & Rashidi, P. (2016). Deep neural network architectures for forecasting analgesic response. *Engineering in Medicine and Biology Society*.

Pedregosa, F. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*.

Pham, T., Tran, T., Phung, D., & Venkatesh, S. (2017). DeepCare: A Deep Dynamic Memory Model for Predictive Medicine. *arXiv:1602.00357*.

Purushotham, S., Meng, C., Che, Z., & Liu, Y. (2017). Benchmark of Deep Learning Models on Large Healthcare MIMIC Datasets. *Journal of Biomedical Informatics*.

Shin, J. (2015). Incremental knowledge base construction using deepdive. *Proceedings of the VLDB Endowment*.

Singh, H., Meyer, A., & Thomas, E. (2014). The frequency of diagnostic errors in outpatient care: estimations from three large observational studies involving US adult populations. *BMJ Quality & Safety*.

Song, H., Rajan, D., & Spanias, A. (2018). Attend and Diagnose: Clinical Time Series Analysis using Attention Models. *AAAI*.

Tran, T., Nguyen, T. D., Phung, D., & Venkatesh, S. (2015). Learning vector representation of medical objects via EMR-driven nonnegative restricted Boltzmann machines. *Journal of Biomedical Informatics*.

Winters, B., Custer, J., & Newman-Toker, D. (2012). Diagnostic errors in the intensive care unit: a systematic review of autopsy studies. *BMJ Quality Safety*.

Wu, Y. (2015). Named entity recognition in Chinese clinical text using deep neural network. *Studies in health technology and informatics*.

Xiao, C., Choi, E., & Sun, J. (2018). Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review. *Journal of the American Medical Informatics Association*.